Mini Project Report

on

IMPLEMENTATION OF FIVE FACTOR MODEL USING MACHINE LEARNING

By

PRABIN DEWAN (202100364)
HANSIKA SHYAMALA (202100470)
ADARSH VERMA (202100541)
DEVANSH TYAGI (202100542)
ANIKET KUMAR (202100560)
Group Id – C15

In partial fulfillment of requirements for the award of degree in

Bachelor of Technology in Computer Science and Engineering

(2024)

Under the Project Guidance of

Mrs. Minakshi Roy

Assistant Professor (SG), Dept. of CSE

Sikkim Manipal Institute of Technology, Majitar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY

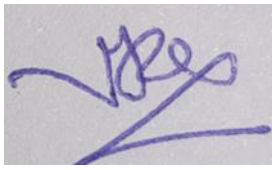
(A constituent college of Sikkim Manipal University)

MAJITAR, RANGPO, EAST SIKKIM – 737136

PROJECT COMPLETION CERTIFICATE

This is to certify that the below mentioned students of Sikkim Manipal Institute of Technology have worked under my supervision and guidance from **8th January 2024 to 17th April 2024** and successfully completed the Mini project entitled **"IMPLEMENTATION OF FIVE FACTOR MODEL USING MACHINE LEARNING"** in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Course
202100364	**Prabin Dewan**	**B.Tech (CSE)**
202100470	**Hansika Shyamala**	**B.Tech (CSE)**
202100541	**Adarsh Verma**	**B.Tech (CSE)**
202100542	**Devansh Tyagi**	**B.Tech (CSE)**
202100560	**Aniket Kumar**	**B.Tech (CSE)**



Mrs. Minakshi Roy

Assistant Professor (SG)

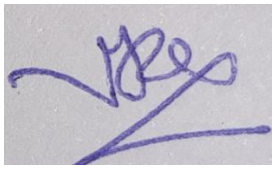
Department of Computer Science and Engineering

Sikkim Manipal institute of Technology

Majhitar, Sikkim – 737136

PROJECT REVIEW CERTIFICATE

This is to certify that the work recorded in this project report entitled **"IMPLEMENTATION OF FIVE FACTOR MODEL USING MACHINE LEARNING"** has been jointly carried out by **Prabin Dewan (Reg. 202100364), Hansika Shyamala (Reg. 202100470), Adarsh Verma (Reg. 202100541), Devansh Tyagi (Reg. 202100542) and Aniket Kumar (Reg. 202100560)** of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Mini Project Viva Examination.



Mrs. Minakshi Roy

Assistant Professor (SG)

Department of Computer Science and Engineering

Sikkim Manipal Institute of Technology

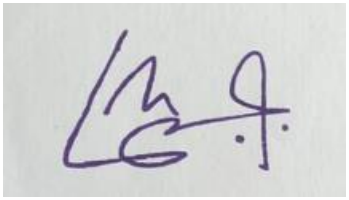
Majhitar, Sikkim – 737136

CERTIFICATE OF ACCEPTANCE

This is to certify that the below mentioned students of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) have worked under the supervision of **Mrs. Minakshi Roy**, Assistant Professor, Department of Computer Science and Engineering from **8th January 2024 to 17th April 2024** on the project entitled **"IMPLEMENTATION OF FIVE FACTOR MODEL USING MACHINE LEARNING".**

The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Project Venue
202100364	**Prabin Dewan**	
202100470	**Hansika Shyamala**	
202100541	**Adarsh Verma**	**SMIT**
202100542	**Devansh Tyagi**	
202100560	**Aniket Kumar**	



Dr. Udit Kumar Chakraborty

Professor & Head of the Department

Computer Science & Engineering Department

Sikkim Manipal Institute of Technology

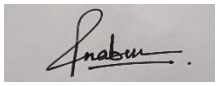
Majhitar, Sikkim – 737136

DECLARATION

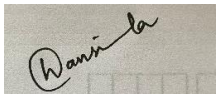
We, the undersigned, hereby declare that the work recorded in this project report entitled **"IMPLEMENTATION OF FIVE FACTOR MODEL USING MACHINE LEARNING"** in partial fulfillment for the requirements of award of B.Tech (CSE) from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bona fide project work carried out at "**SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY**" under the supervision and guidance of **Mrs. Minakshi Roy,** Assistant Professor (SG), Department of Computer Science and Engineering.

The results of this investigation reported in this project have so far not been reported for any other Degree or any other Technical forum.

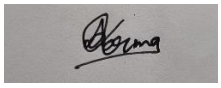
The assistance and help received during the course of the investigation have been duly acknowledged.



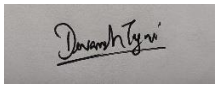
Prabin Dewan (Reg. No.-202100364)



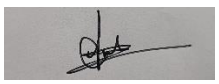
Hansika Shyamala (Reg. No.-202100470)



Adarsh Verma (Reg. No.-202100541)



Devansh Tyagi (Reg. No.-202100542)



Aniket Kumar (Reg. No.-202100560)

ACKNOWLEDGMENT

We take this opportunity to acknowledge indebtedness and a deep sense of gratitude to our guide **Mrs. Minakshi Roy** for her valuable guidance and supervision throughout the course which shaped the present work as it shows.

We pay our deep sense of gratitude to **Prof. (Dr.) Udit Kumar Chakraborty, HOD, Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us the opportunity to work on this project and providing all support required.

We are obliged to our Mini Project coordinators **Dr. Sandeep Gurung, Mr. Biraj Upadhyaya** and **Ms. Tanuja Subba** for elevating, inspiration and supervising in completion of our project.

We would also like to thank any other staff of **Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us continuous support and guidance that has helped us in completion of our project.

Prabin Dewan (Reg. No.-202100364)

Hansika Shyamala (Reg. No.-202100470)

Adarsh Verma (Reg. No.-202100541)

Devansh Tyagi (Reg. No.-202100542)

Aniket Kumar (Reg. No.-202100560)

DOCUMENT CONTROL SHEET

1	Report No	CSE/Mini Project/Internal/B.Tech/Section/Group ID: C15/2024
2	Title of the Report	Implementation of Five Factor Model Using Machine Learning
3	Type of Report	Technical
4	Author	Prabin Dewan, Hansika Shyamala, Adarsh Verma, Devansh Tyagi and Aniket Kumar
5	Organizing Unit	Sikkim Manipal Institute of Technology
6	Language of the Document	English
7	Abstract	This project explores the application of machine learning techniques, specifically clustering, to analyze responses from a questionnaire aimed at assessing five personality traits: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism (OCEAN). The dataset consists of responses collected from participants who completed the questionnaire, providing insights into their individual personalities.
8	Security Classification	General
9	Distribution Statement	General

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This project aims to leverage machine learning, specifically clustering algorithms, to gain insights from a dataset comprising responses to a questionnaire designed to assess five key personality traits known as OCEAN: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. By analysing these responses, we endeavour to uncover underlying patterns and associations within the data, shedding light on the diverse spectrum of human personalities. To begin, the dataset undergoes rigorous preprocessing steps to ensure its quality and consistency. Once cleaned, clustering algorithms such as K-Means or Hierarchical Clustering are applied to segment the dataset into distinct groups based on similarities in participants' responses. Through this process, individuals with similar personality profiles are grouped together, facilitating the identification of common behavioural tendencies and relationships among the OCEAN traits.

Evaluation metrics and visualization techniques are employed to assess the quality of the clustering results and to aid in the interpretation of findings. The outcomes of this project hold significant implications for various domains, including psychology, sociology, and human-computer interaction. By uncovering patterns in personality traits, we can develop personalized recommendations, targeted interventions, and tailored experiences that cater to individuals' unique characteristics. Ultimately, this research contributes to advancing our understanding of human behaviour and personality, offering valuable insights for applications in diverse real-world contexts.

1. INTRODUCTION

The study of personality, exploring unique mental, emotional, and behavioral traits, has been a longstanding area of interest. Enter the Big Five personality model – a framework that looks at five key traits: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. In this project, we delve into predicting individual personalities using machine learning, particularly focusing on the Big Five model.

In a world where machine learning touches many aspects of our lives, understanding how personality and computational models connect becomes intriguing. Our goal is to use machine learning to predict a person's personality, concentrating on the Big Five model. We will be looking at collaborative filtering, a widely used recommendation engine, as the framework for our exploration.

Unlike studies in specific domains, our project takes a broader view, exploring how incorporating personality traits into collaborative filtering can impact various areas. This initiative not only deepens our understanding of the interplay between personality and machine learning but also has the potential to improve recommendation systems on a larger scale. As we embark on this journey, the blend of psychology and machine learning unfolds, offering a promising path to uncover hidden patterns in the complex landscape of individual differences.

2. LITERATURE SURVEY

Sl. No.	Author	Paper and Publication Details	Findings	Relevance
1	Tejas Bakade, Siddhant Deshpande, Omkar Pethe, Shreyas Kale, Yashwant Madane, P. D. J. (2022)	Big Five Personality Prediction Using Machine Learning Algorithms. Mathematical Statistician and Engineering Applications, 71(3), 1128 – . https://doi.org/10.17762/msea.v71i3.393	Using different machine learning techniques and algorithms, it has been observed that predicting personality is possible. The different techniques used were ML Decision Trees, neural networks and K-means clustering, etc.	The mentioned research findings provide a foundation and guidance for your project on implementing the Big Five Personality Model using machine learning. They help justify the choice of methods, provide benchmarks for evaluation, and inform considerations regarding data and potential future developments in the field.
2	Chi, J., & Chi, Y. N. (2023)	Cluster Analysis of Personality Types Using Respondents' Big Five Personality Traits. International Journal of Data Science, 4(2), 116-135	In this study, a k-means clustering analysis was applied, as a result, a four-cluster solution was identified, which was labelled as Role Models, Average, Reserved, And Self-Centred clusters, representing distinct personality types.	In our project, we will draw insights from this approach to potentially identify clusters representing different levels or combinations of the Big Five traits.

Sl. No.	Author	Paper and Publication Details	Findings	Relevance
3	Johnson, K., Khaire, S., Nage, Y., & Kaul P.	PERSONALITY PREDICTION APP USING BIG-FIVE TRAITS	The research compared different model like Logistic Regression, Random Forest, Support Vector Machine, and Nave Bayes Classifier. The model showing the most accuracy was the Logistics Regression Model with 85.71 % accuracy.	We can consider Logistic Regression as a baseline or primary model for predicting the Big Five personality traits. The reported accuracy of 85.71% serves as a benchmark for evaluating the performance of your Big Five Personality Model. We will use this accuracy metric to assess the effectiveness of your model and strive to achieve comparable or improved results.

3. PROBLEM DEFINITION

- The accuracy of current working models is very low to be actually used in real world applications. The primary focus of this project is to overcome the accuracy of previous models and make it precise enough to be used in practical scenarios.
- The goal of this project is to analyze the working of current models, find their shortcomings and then create a new model which is more accurate and can be applied in real world applications.
- The clusters formed from the dataset have a lot of overlapping, and distinct clusters are not able to be formed due to which the prediction accuracy is low.
- The number of features is a lot which increases the computation for training machine learning models. This can lead to sparsity in the data and make it harder for algorithms to find meaningful patterns, as the data becomes more spread out.

4. SOLUTION STRATEGY

Step 1: Data Preprocessing and Cleaning

In this step, the raw training data is processed and cleaned to ensure its quality and suitability for machine learning tasks. This typically involves tasks such as:

- Handling missing values: Identifying and addressing missing values in the dataset, either by imputing them with a suitable value or removing the corresponding records.

- Data normalization or standardization: Scaling numerical features to a standard range to prevent features with larger magnitudes from dominating the model training process.

- Removing outliers: Identifying and handling outliers that might skew the model's performance or accuracy.

Data preprocessing is crucial as it helps improve the quality of input data and enhances the performance of machine learning models.

Step 2: Exploratory Data Analysis (EDA)

EDA involves analyzing preprocessed data using various statistical and visualization techniques to gain insights and understanding of its underlying structure. Common tasks performed during EDA include:

- Summary statistics: Computing descriptive statistics such as mean, median, standard deviation, and quartiles to summarize the distribution of numerical features.

- Data visualization: Creating visualizations such as histograms, scatter plots, and box plots to visualize the distribution, relationships, and patterns within the data.

- Correlation analysis: Examining the correlations between different features in the dataset to identify potential relationships or dependencies.

EDA helps data scientists and analysts uncover patterns, trends, and anomalies in the data, which can inform feature engineering and model selection decisions.

Step 3: Model Training

In this step, machine learning models such as K-means clustering, and Self-Organizing Maps (SOM) are trained using the preprocessed data. Each model is trained on the training data and evaluated on validation data to assess its performance. Key aspects of this step include:

- Model selection: Choosing appropriate machine learning algorithms or models based on the nature of the problem and the characteristics of the data.

Step 4: Model Evaluation

In this step, various evaluation metrics such as the Elbow method and silhouette score are used to evaluate the performance of clustering models (K-means and SOM). These metrics help assess the quality and effectiveness of the clustering algorithms in partitioning the data into meaningful clusters. For example:

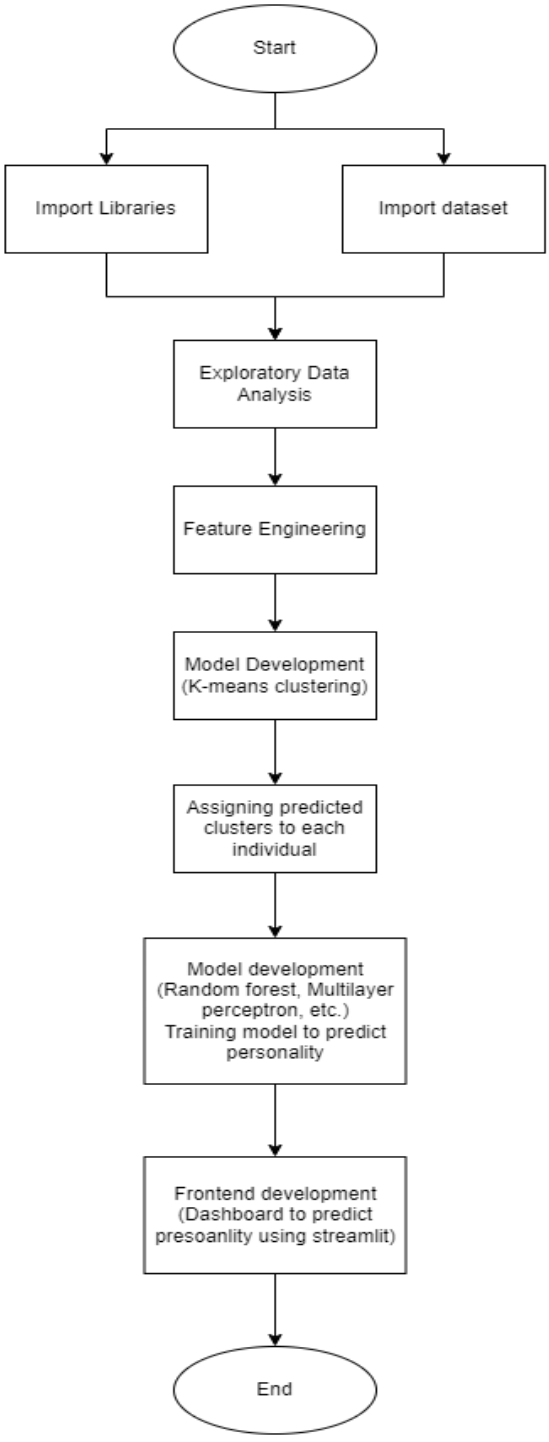
- The Elbow method helps determine the optimal number of clusters by plotting the within-cluster sum of squares (WCSS) against the number of clusters and identifying the "elbow" point where the rate of decrease in WCSS slows down.

By evaluating clustering models using appropriate metrics, we can objectively assess their performance and choose the most suitable model for the task at hand.

Overall, these steps constitute a structured approach to preprocessing, analyzing, and modeling data in our project.

5. DESIGN

5.1. Flow chart



6. IMPLEMENTATION DETAILS

6.1. Algorithms

K-Means clustering:

K-Means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into K distinct, non-overlapping clusters. The algorithm iteratively assigns data points to the nearest cluster centroid and then updates the centroids based on the mean of the data points assigned to each cluster. This process continues until convergence, where centroids no longer change significantly, or a specified number of iterations is reached. K-Means aims to minimize the within-cluster variance, making data points within the same cluster as similar as possible and maximizing the between-cluster variance. It's widely used for various tasks, including customer segmentation, image compression, and anomaly detection, due to its simplicity, efficiency, and effectiveness in identifying patterns in data.

```
df_model_4 = df.drop(['race', 'age', 'engnat', 'gender', 'hand', 'source', 'country'], axis=1)

kmeans = KMeans(n_clusters=4, random_state=42)
k_fit = kmeans.fit(df_model_4)

predictions = k_fit.labels_
df_model_4['clusters'] = predictions
```

SOM (Self-Organizing Maps):

Self-Organizing Maps (SOM) is an unsupervised machine learning algorithm used for dimensionality reduction and visualization of high-dimensional data. It involves creating a grid of nodes or neurons arranged in a lattice structure, where each neuron represents a prototype of the input data. During training, SOM iteratively adjusts the weights of these neurons to best match the input data distribution. Neurons closer to each other in the grid tend to represent similar data points, allowing SOM to preserve the topological relationships present in the input data. SOM is often used for exploratory data analysis, clustering, and visualizing complex datasets, particularly in cases where the underlying structure of the data is not well understood.

```
!pip install minisom
from minisom import MiniSom

# Train SOM
som_shape = (5, 5)  # Adjust grid size as needed
som = MiniSom(som_shape[0], som_shape[1], data_scaled.shape[1], sigma=0.5, learning_rate=0.5)
som.train_random(data_scaled, 1000000)  # Adjust number of iterations as needed
```

Random forest:

Random Forest is a versatile and powerful supervised learning algorithm used for classification and regression tasks. It operates by constructing multiple decision trees during training and outputs the mode (classification) or average (regression) prediction of the individual trees. Each decision tree in the Random Forest is trained on a bootstrap sample of the original data, and at each split, a random subset of features is considered. This randomness helps to decorrelate the individual trees and improve the overall performance of the ensemble model. Random Forest is robust to overfitting, handles high-dimensional data well, and provides built-in feature importance measures. It's widely used across various domains due to its high accuracy, scalability, and ease of use.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

data = pd.read_csv('final.csv')

X = data.drop(columns=['Predictions'])
y = data['Predictions']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

rf_classifier.fit(X_train, y_train)

y_pred = rf_classifier.predict(X_test)
```

ADA Boost:

AdaBoost, short for Adaptive Boosting, is a popular ensemble learning method used for classification tasks. It operates by combining multiple weak learners, typically decision trees, to create a strong classifier. During training, AdaBoost assigns higher weights to misclassified data points, allowing subsequent weak learners to focus more on the difficult-to-classify instances. Each weak learner is trained sequentially, and at each iteration, the algorithm adjusts the weights of the training samples to emphasize the misclassified points from the previous iteration. By iteratively combining weak learners, AdaBoost creates a robust ensemble model that achieves high accuracy even with simple base classifiers. It's known for its ability to handle imbalanced datasets, reduce bias and variance, and improve classification performance compared to individual classifiers.

```
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X = data.drop(columns=['Predictions']).values
y = data['Predictions']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

adaboost_clf = AdaBoostClassifier(n_estimators=50, random_state=42)

adaboost_clf.fit(X_train, y_train)

y_pred = adaboost_clf.predict(X_test)
```

XG Boost:

XGBoost (Extreme Gradient Boosting) is a powerful and efficient implementation of gradient boosting, a machine learning technique used for supervised learning tasks such as classification, regression, and ranking. XGBoost builds an ensemble of weak learners, typically decision trees, in a sequential manner, where each new tree corrects the errors made by the previous ones. During training, XGBoost optimizes a differentiable loss function by iteratively adding new trees to minimize the overall prediction error. It incorporates regularization techniques to prevent overfitting and parallel processing to enhance scalability and efficiency. XGBoost is renowned for its high performance, accuracy, and versatility, making it a popular choice for various machine learning competitions and real-world applications.

```
import pandas as pd
from sklearn.model_selection import train_test_split
import xgboost as xgb
from sklearn.metrics import accuracy_score

data = pd.read_csv('final.csv')

X = data.drop(columns=['Predictions'])
y = data['Predictions']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = xgb.XGBClassifier(objective='multi:softmax', num_class=len(y.unique()))

model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

Naïve bayes:

Naive Bayes is a probabilistic machine learning algorithm commonly used for classification tasks. It's based on Bayes' theorem, which calculates the probability of a hypothesis given the evidence. Naive Bayes assumes that the features are conditionally independent given the class label, hence the "naive" assumption. Despite this simplification, Naive Bayes often performs well, especially in text classification and spam filtering tasks. It's computationally efficient, requires a small amount of training data, and is robust to irrelevant features. Naive Bayes is suitable for binary and multi-class classification problems and provides a straightforward interpretation of results.

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

nb_classifier = GaussianNB()

nb_classifier.fit(X_train, y_train)

y_pred = nb_classifier.predict(X_test)
```

Multilayer perceptron:

Multilayer Perceptron (MLP) is a type of artificial neural network composed of multiple layers of interconnected neurons. It consists of an input layer, one or more

hidden layers, and an output layer. Each neuron in the hidden layers uses an activation function to transform the weighted sum of its inputs, allowing the network to learn complex, non-linear relationships in the data. During training, MLP employs backpropagation to update the weights of the connections between neurons, minimizing the difference between the predicted and actual outputs. MLPs can approximate any continuous function, making them versatile for various supervised learning tasks such as classification, regression, and even pattern recognition. They can handle large and complex datasets but may require careful tuning of hyperparameters to prevent overfitting.

```
import pandas as pd
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

data = pd.read_csv('final.csv')

X = data.drop(columns=['Predictions'])
y = data['Predictions']

y = pd.get_dummies(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(64, activation='relu'),
    Dense(y.shape[1], activation='softmax')
])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy','mean_absolute_error'])

model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_test, y_test))
```

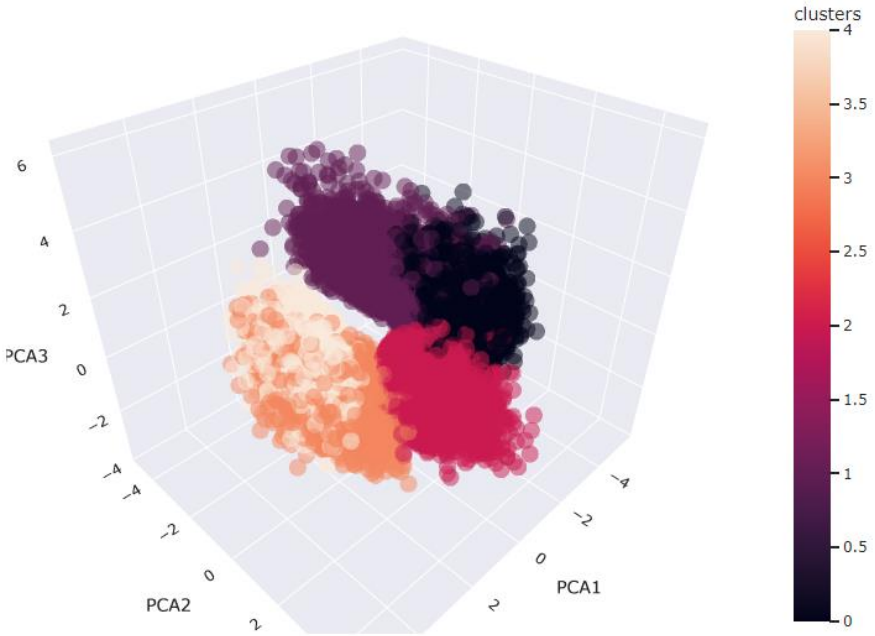
6.2. Details of dataset

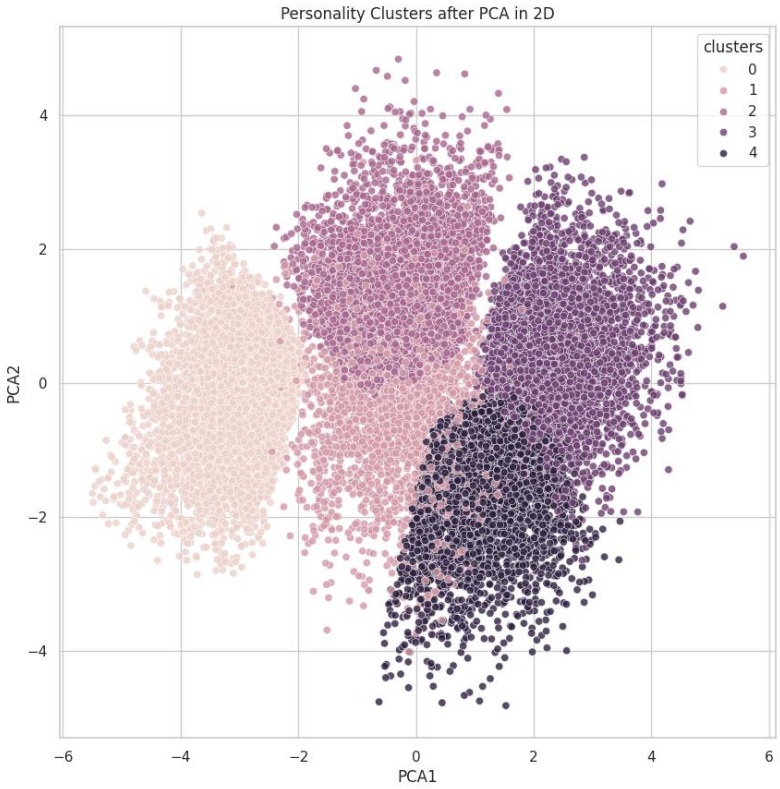
Name of model	No. of rows (size)	Type of Data	How data were acquired	Parameters for data collection	Description of Data collection	Data Accessibility
Five factor model (OCEAN Model)	19,706 (4.3 MB)	Integer, String	This dataset contains 19,706 questionnaire answers collected online by OPEN Psychometrics.	• Extraversion • Agreeableness • Conscientiousness • Neuroticism • Openness to Experiences	Learners answer statements with a choice of strongly disagree, Disagree, doubt, agree and Strongly agree, scores range from 1,2,3,4 and 5.	https://www.kaggle.com/code/kattat/four-clusters-analysis-from-big5-personality-test/input?select=data.csv

7. RESULTS AND DISCUSSION

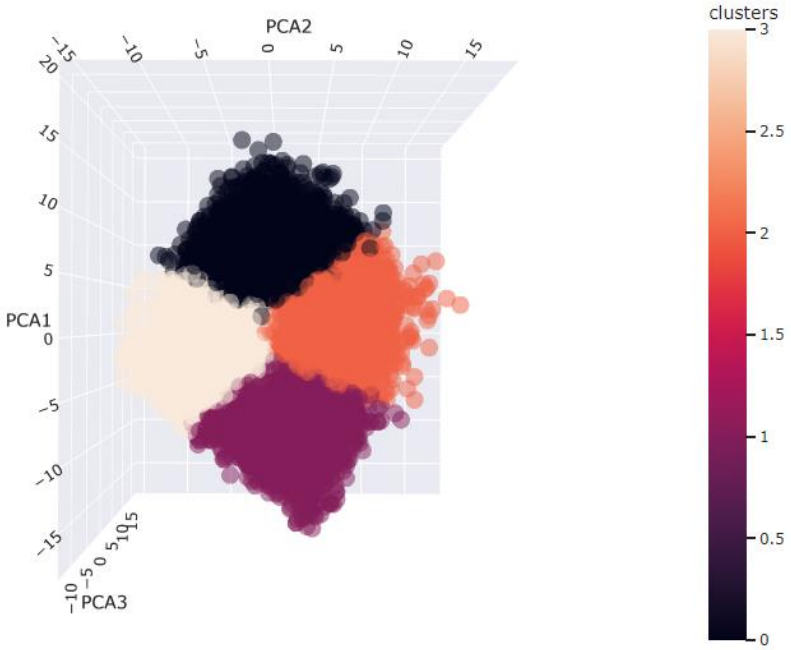
7.1. K-Means clustering

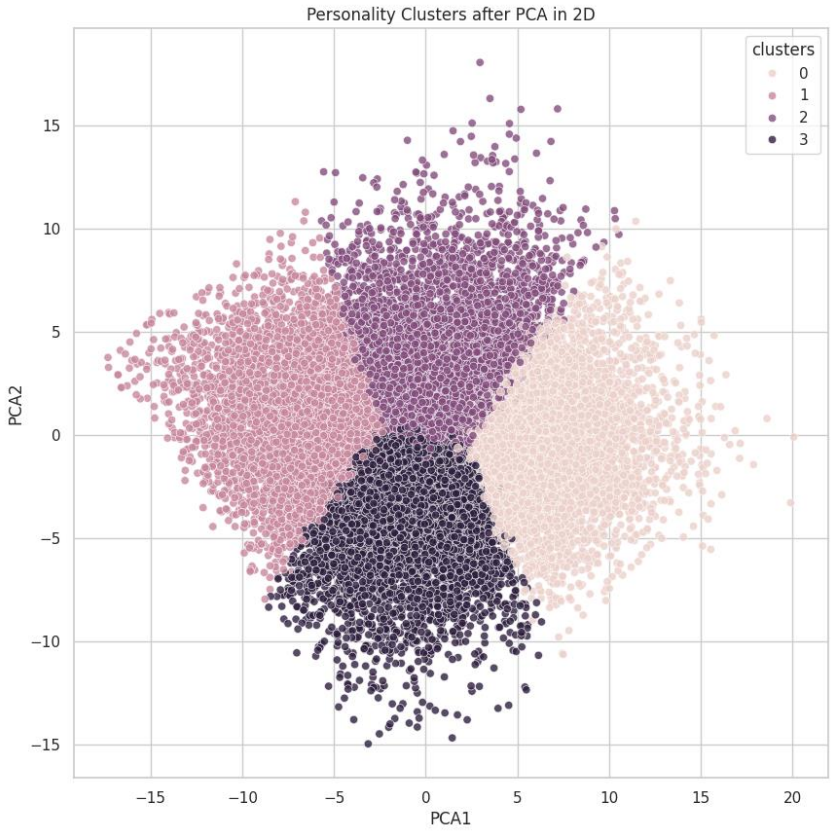
1. **Before feature engineering**





2. After feature engineering





This plot really nicely shows how the 4 clusters differ from each other.

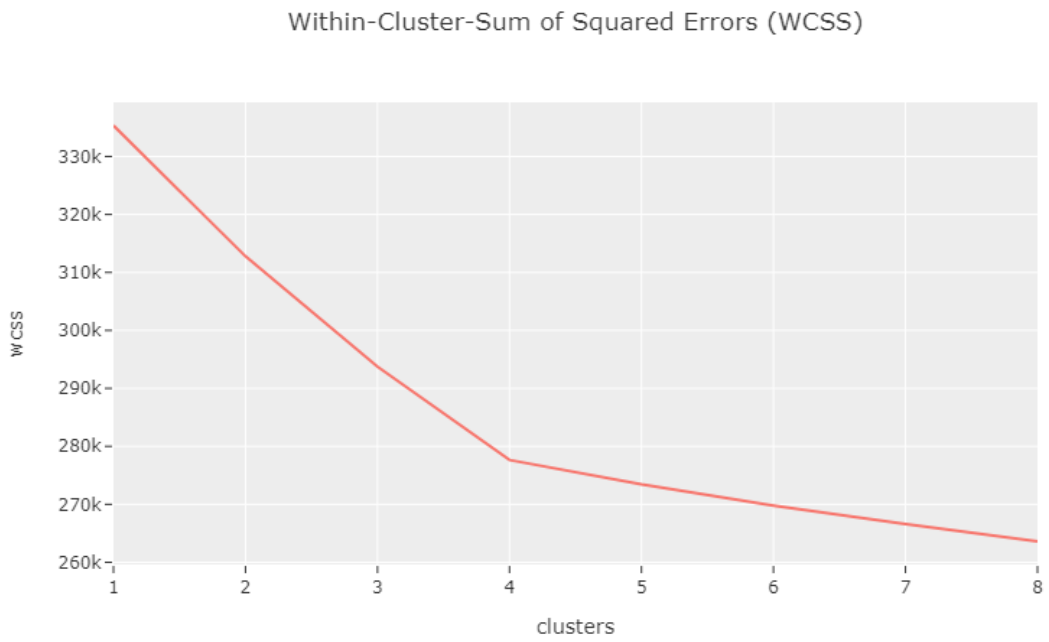
- First cluster is made of most introverted, quite nervous, friendly and creative but neither careless nor organized personalities.
- Second cluster is made of most extrovert, friendly, organized and curious and also most confident people.
- Third cluster can be described as consisting of mid-level confident introverts who are rather friendly, organized and quite creative.
- And the last cluster consists of mid-level nervous extraverts who are really friendly and creative but neither careless nor organized.

3. Elbow analysis

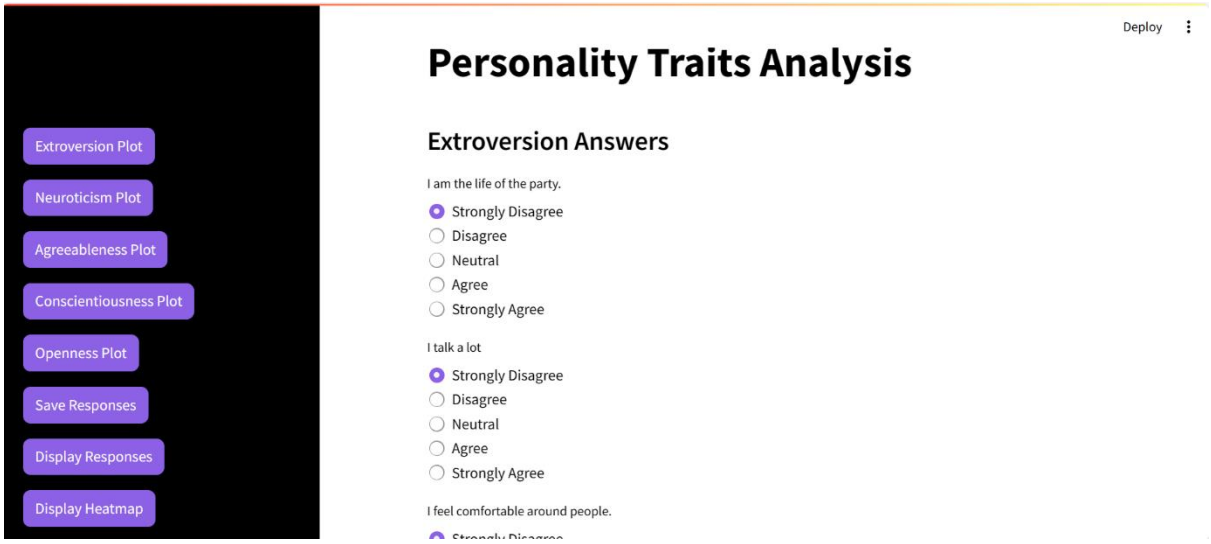
```
wcss = []
for i in range(2, 10):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(df_model)
    wcss.append(kmeans.inertia_)

wcss = pd.DataFrame(wcss, columns=['wcss'])
wcss = wcss.reset_index()
wcss = wcss.rename(columns={'index': 'clusters'})
wcss['clusters'] += 1
wcss.head()
```

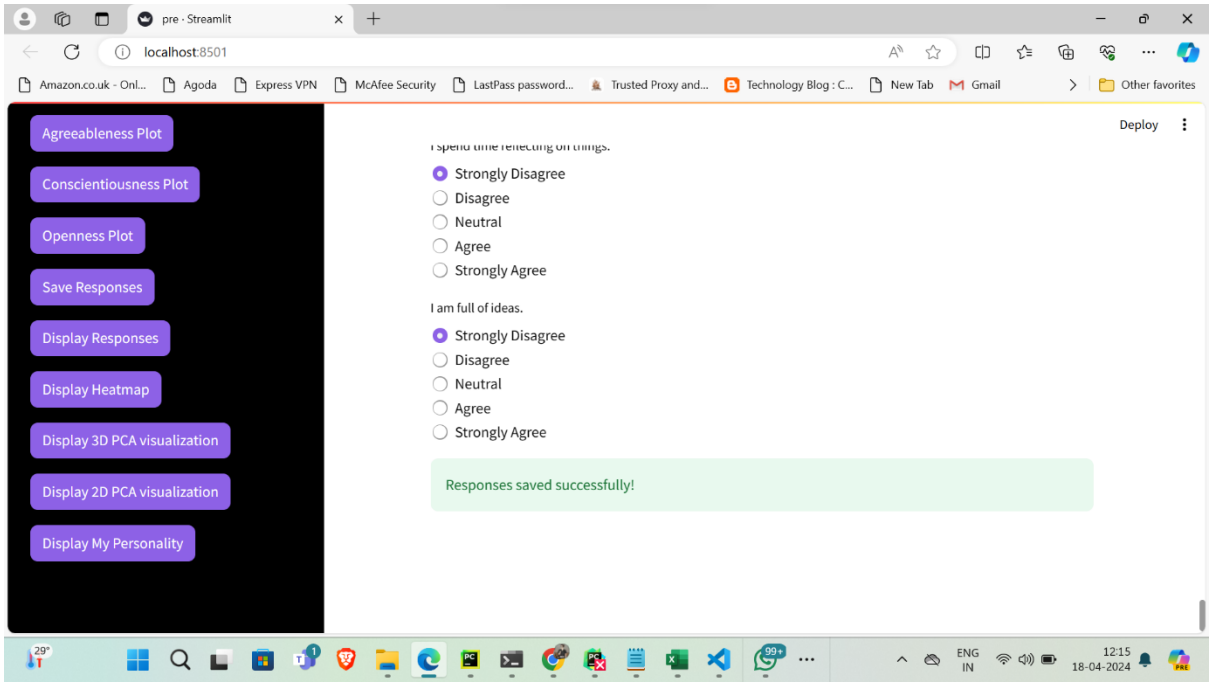
	clusters	wcss
0	1	335342.971582
1	2	312777.191230
2	3	293738.228457
3	4	277650.818155
4	5	273446.896208

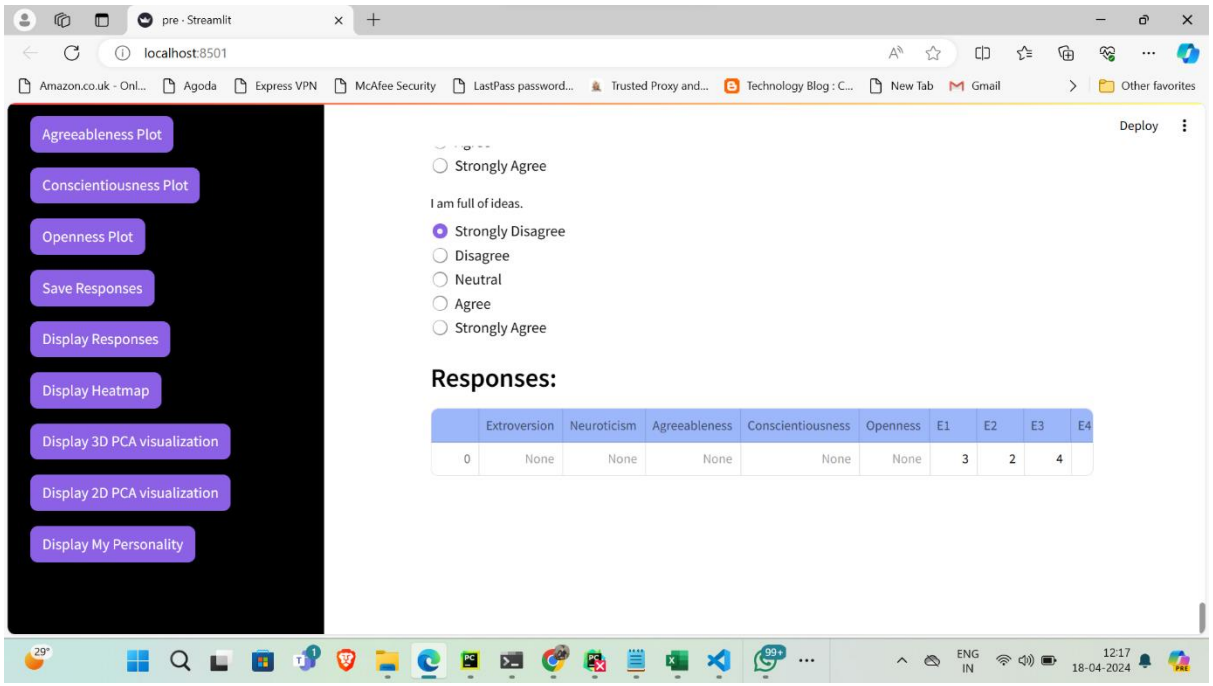
```
px.line(wcss, x='clusters', y='wcss', width=800, height=500, title='Within-Cluster-Sum of Squared Errors (WCSS)',
        template='ggplot2')
```

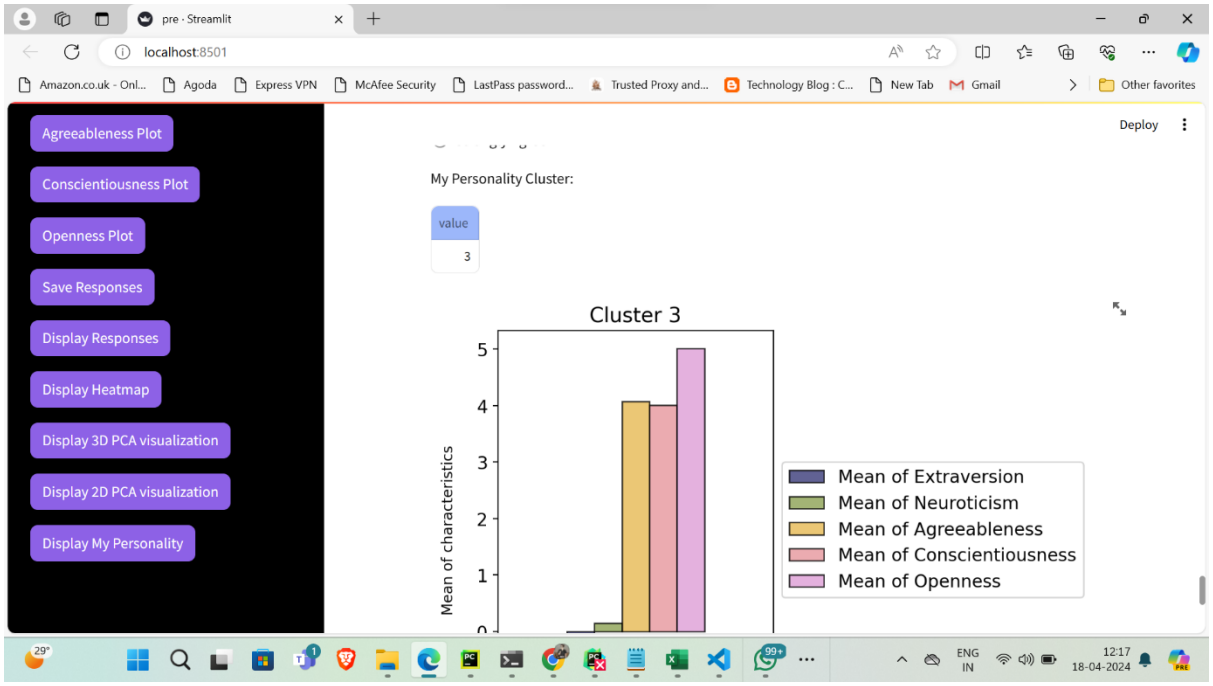


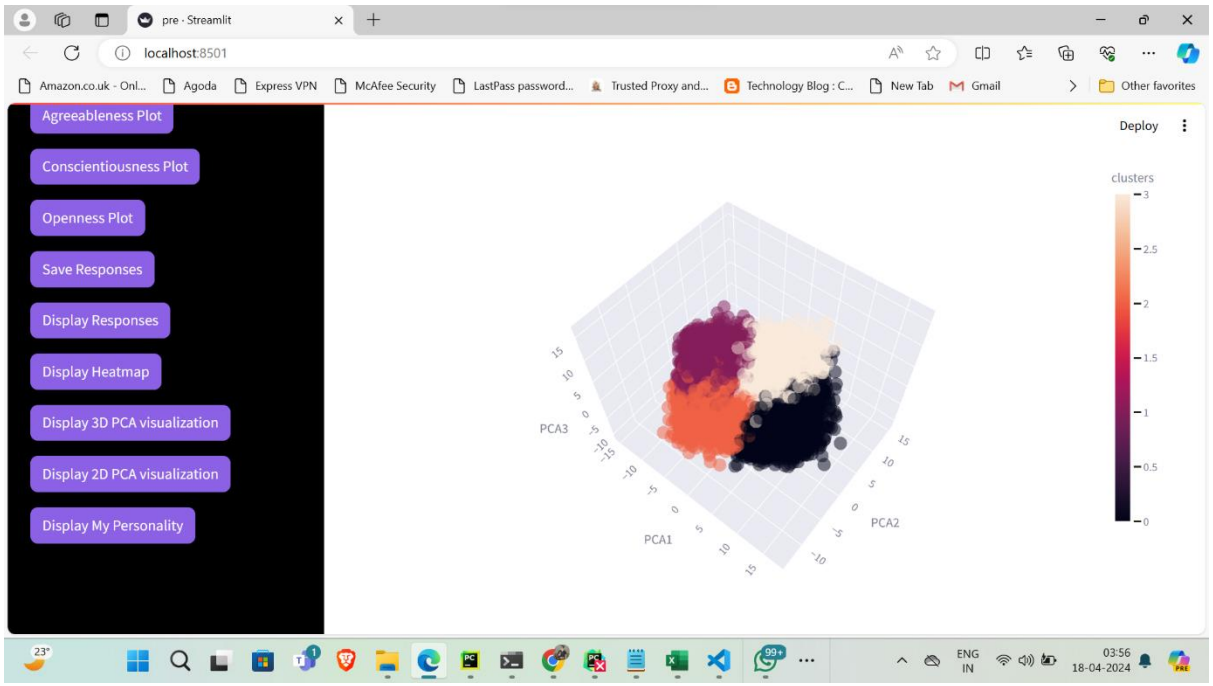
4. Frontend Development

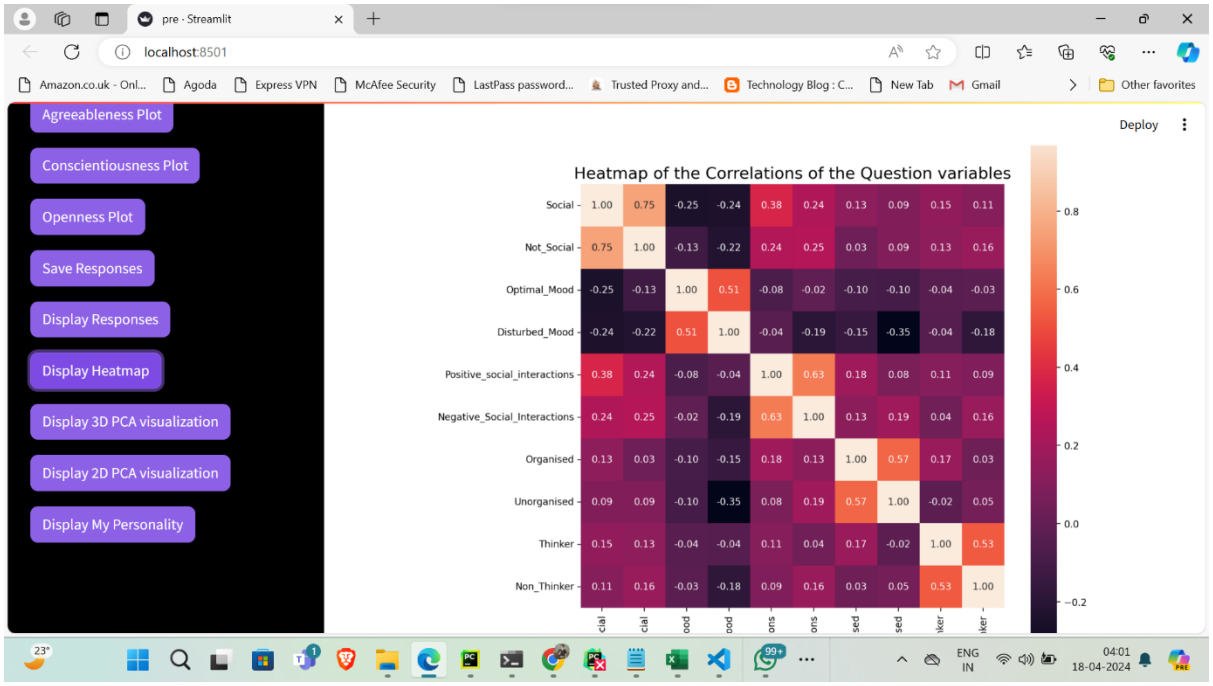



pre · Streamlit
localhost:8501
Agreeableness Plot
Conscientiousness Plot
Openness Plot
Save Responses
Display Responses
Display Heatmap
Display 3D PCA visualization
Display 2D PCA visualization
Display My Personality
Deploy
Strongly Disagree
Disagree
Neutral
Agree
Strongly Agree
I am full of ideas.
Strongly Disagree
Disagree
Neutral
Agree
Strongly Agree
Responses saved successfully!


pre · Streamlit
localhost:8501
Agreeableness Plot
Conscientiousness Plot
Openness Plot
Save Responses
Display Responses
Display Heatmap
Display 3D PCA visualization
Display 2D PCA visualization
Display My Personality
Deploy
Strongly Agree
I am full of ideas.
Strongly Disagree
Disagree
Neutral
Agree
Strongly Agree
Responses:
Extroversion
Neuroticism
Agreeableness
Conscientiousness
Openness
E1
E2
E3
E4
0
None
None
None
None
None
3
2
4


pre · Streamlit
localhost:8501
Agreeableness Plot
Conscientiousness Plot
Openness Plot
Save Responses
Display Responses
Display Heatmap
Display 3D PCA visualization
Display 2D PCA visualization
Display My Personality
Deploy
My Personality Cluster:
value
3
Cluster 3
Mean of characteristics
Mean of Extraversion
Mean of Neuroticism
Mean of Agreeableness
Mean of Conscientiousness
Mean of Openness


pre · Streamlit
localhost:8501
Agreeableness Plot
Conscientiousness Plot
Openness Plot
Save Responses
Display Responses
Display Heatmap
Display 3D PCA visualization
Display 2D PCA visualization
Display My Personality
Deploy
clusters
PCA1
PCA2
PCA3


pre · Streamlit
localhost:8501
Amazon.co.uk - Onl...
Agoda
Express VPN
McAfee Security
LastPass password...
Trusted Proxy and...
Technology Blog : C...
New Tab
Gmail
Other favorites
Agreeableness Plot
Conscientiousness Plot
Openness Plot
Save Responses
Display Responses
Display Heatmap
Display 3D PCA visualization
Display 2D PCA visualization
Display My Personality
Deploy
Heatmap of the Correlations of the Question variables
Social 1.00 0.75 -0.25 -0.24 0.38 0.24 0.13 0.09 0.15 0.11
Not_Social 0.75 1.00 -0.13 -0.22 0.24 0.25 0.03 0.09 0.13 0.16
Optimal_Mood -0.25 -0.13 1.00 0.51 -0.08 -0.02 -0.10 -0.10 -0.04 -0.03
Disturbed_Mood -0.24 -0.22 0.51 1.00 -0.04 -0.19 -0.15 -0.35 -0.04 -0.18
Positive_social_interactions 0.38 0.24 -0.08 -0.04 1.00 0.63 0.18 0.08 0.11 0.09
Negative_Social_Interactions 0.24 0.25 -0.02 -0.19 0.63 1.00 0.13 0.19 0.04 0.16
Organised 0.13 0.03 -0.10 -0.15 0.18 0.13 1.00 0.57 0.17 0.03
Unorganised 0.09 0.09 -0.10 -0.35 0.08 0.19 0.57 1.00 -0.02 0.05
Thinker 0.15 0.13 -0.04 -0.04 0.11 0.04 0.17 -0.02 1.00 0.53
Non_Thinker 0.11 0.16 -0.03 -0.18 0.09 0.16 0.03 0.05 0.53 1.00
0.8
0.6
0.4
0.2
0.0
−0.2
ENG
IN
04:01
18-04-2024

8. CONCLUSION AND FUTURE SCOPE

To wrap up, this study looked at how we can group people based on the Big Five Personality Traits using smart methods like discriminant analysis and MLP neural networks. The super accurate results show how well these techniques work in understanding how people's personalities are made up.

The study found that two traits, being worried a lot (neuroticism) and being organized and careful (conscientiousness), are really important in how people see themselves. This shows just how much these traits affect how we behave.

Also, the study shows how knowing about different personality types can help in workplaces. By understanding and adapting to how people are different, companies can improve how they lead, talk to each other, and work together.

In short, this research helps us understand personality types better and how they affect real-life situations. It reminds us that everyone is different, and that's important to remember, especially in workplaces and other settings where people interact.

Looking forward, there's a lot more we can explore and do with the findings from this study. We can dig deeper into understanding how different things besides just the Big Five traits affect how people act. With better technology and analysis methods, we might be able to create more personalized ways to understand and help people based on their personalities. This could be useful in areas like marketing or mental health support. In short, there's a lot of potential for using what we've learned to make life better for everyone.

9. GANTT CHART

ACTIVITY	TIME FRAME			
	January 2024	February 2024	March 2024	April 2024
Literature Survey				
Problem Definition				
Design and Development				
System Analysis and Testing				
Documentation and Report Writing				

Achieved activity **Ongoing activity** **Proposed activity**

10. REFERENCES

1. Tejas Bakade, Siddhant Deshpande, Omkar Pethe, Shreyas Kale, Yashwant Madane, P. D. J. (2022). Big Five Personality Prediction Using Machine Learning Algorithms. Mathematical Statistician and Engineering Applications, 71(3), 1128 –. https://doi.org/10.17762/msea.v71i3.393
2. Chi, J., & Chi, Y. N. (2023). Cluster Analysis of Personality Types Using Respondents' Big Five Personality Traits. International Journal of Data Science, 4(2), 116-135.
3. Johnson, K., Khaire, S., Nage, Y., & Kaul (2022), P. PERSONALITY PREDICTION APP USING BIG-FIVE TRAITS.

11. PLAGIARISM REPORT

Title: IMPLEMENTATION OF FIVE FACTOR MODEL USING MACHINE LEARNING

Submitted by: Prabin Dewan, Hansika Shyamala, Adarsh Verma, Devansh Tyagi and Aniket Kumar

Date of Submission: 18-04-2024

Summary: We're exploring how machine learning can predict people's personalities, focusing on the Big Five model – a framework that looks at five key traits: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism.

In a world where machine learning is everywhere, we're curious about the link between personality and computational models. Our goal is to use machine learning to guess someone's personality traits, with a spotlight on the Big Five model.

Plagiarism Detection Results: Upon analysis using plagiarism detection software, no instances of plagiarism were identified in the submitted report. The content appears to be original and does not closely resemble any existing sources.

Recommendations: Given that no instances of plagiarism were detected, no corrective actions are necessary. The submitted report demonstrates originality and adherence to academic integrity standards.